Jurnal InFact Sains dan Komputer
Vol.8, No.02, Juli 2024
p-ISSN : 2527-8363, e-ISSN: 2829-0259
DOI: https://doi.org/10.61179/jurnalinfact.v8i02.586

Aplikasi Verifikasi Sertifikat Berbasis Website Menggunakan Blockchain

Heri Wijayanto[1*], Pangeran Muhammad Waliyullah[1]
[1]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
[2,]Program Studi Pendidikan Teknologi Informasi, Universitas Muhammadiyah Jakarta

E-mail: heri@unram.ac.id, pangeranwaliyullah@gmail.com

Abstrak

Sertifikat merupakan bukti tercetak atau tertulis dari sebuah lembaga atau individu yang berwenang yang digunakan sebagai bukti kepemilikian atau suatu peristiwa. Seiring dengan mulai banyaknya pemalsuan suatu sertifikat, diperlukannya suatu cara untuk memvalidasi suatu sertifikat tersebut. Maka dari itu dilakukan verifikasi atau permeriksaan tentang kebenaran dari sertifikat tersebut, melakukan verifikasi tentunya dengan cara yang aman dan memiliki kode unik untuk menjaga suatu sertifikat agar sertifikat tidak bisa dipalsukan. Penggunaan blockchain diharapkan dapat menjaga keaslian sertifikat karena didalamnya terdapat kumpulan blok data dengan privasi tinggi, tersimpan di dalam jaringan global yang saling terikat pada sebuah protokol. Sehingga dibuat suatu aplikasi yang dapat membantu validasi tersebut yaitu "Aplikasi Verifikasi Sertifikat Berbasis Website Menggunakan Blockchain" dengan tujuan melakukan verifikasi sertifikat menggunakan blockchain agar sertfikat tersebut dapat dilihat keasliannya serta aman dan tidak dapat dipalsukan.

Info Naskah:
Naskah masuk: 03 Juli 2024
Direvisi: 09 Juli 2024
Diterima: 11 Juli 2024

Abstract

A certificate is printed or written evidence from an authorised institution or individual that is used as proof of ownership or an event. Along with the start of the many forgery of a certificate, a way is needed to validate a certificate. Therefore, verification or examination of the truth of the certificate is carried out, verifying of course in a safe way and has a unique code to maintain a certificate so that the certificate cannot be forged. The use of blockchain is expected to maintain the authenticity of the certificate because it contains a collection of data blocks with high privacy, stored in a global network that is bound together by a protocol. So that an application is made that can help with the validation, namely "Website-Based Certificate Verification Application Using Blockchain" with the aim of verifying certificates using blockchain so that the certificate can be seen for its authenticity and is safe and cannot be forged.

Keywords:
Verifikasi;
Blockchain;
Sertifikat;
Ganache;
Truffle;

***Penulis korespondensi:**
Heri Wijayanto
E-mail: heri@unram.ac.id

1. Pendahuluan

1.1 Latar Belakang

Belakangan ini, teknologi blockchain sering dibahas dan diteliti oleh banyak pihak. Penyebabnya adalah karena blockchain dapat memberikan inovasi baru untuk berbagi dan mendapatkan informasi. Kombinasi teknologi yang dimiliki oleh Blockchain meliputi peer to peer network, smart contract dan mekanisme konsensus, selain kriptografi yang menjadi dasar utama pembuatan blockchain. Selain kriptografi, dasar pembuatan blockchain terdiri dari teknologi seperti mekanisme konsensus, jaringan peer to peer, dan smart contract.

Banyak pihak sedang melakukan sejumlah penelitian untuk melihat potensi penggunaan teknologi blockchain di bidang yang mereka kerjakan. Salah satu kemungkinan pengaplikasian teknologi blockchain yakni aplikasi verifikasi sertifikat. Teknologi telah bergerak cukup maju sampai sekarang. Membedakan sertifikat palsu dan asli akan membutuhkan banyak konsentrasi dan mengakibatkan pemborosan waktu yang berharga. Untuk mengatasi hal tersebut dapat dibuatkan aplikasi verifikasi sertifikat menggunakan teknologi blockchain. Mengapa menggunakan blockchain? Karena data dalam blockchain tidak dapat diubah dalam kondisi realistis. Bahkan jika data tersebut diubah, hanya perlu beberapa saat untuk mengetahui perubahan tersebut. Dalam blockchain data akan divalidasi hanya jika banyak pihak yang menyetujuinya. Jadi sistem akan dapat diandalkan dan diauntentikasi kapan saja.

Karena semua data disimpan secara digital, pengguna tidak perlu khawatir tentang kehilangan sertifikat pada proses validasi.

1.2 Dasar Teori

1.2.1 Blockchain

Dengan teknologi blockchain, data yang dibuat oleh satu server dapat direplikasi dan diverifikasi oleh server lain. Blockchain adalah sistem penyimpanan data digital yang terdiri dari banyak server. Tidak hanya mata uang kripto yang digunakan oleh blockchain. Karena itu, teknologi yang dapat merekam data dan transaksi mini dapat digunakan untuk berbagai hal, seperti dokumentasi, pencatatan proses transaksi, bahkan pemungutan suara [1], [2], [3].

1.2.2 Truffle

Truffle merupakan lingkungan pengembangan kelas dunia, yang menyediakan kerangka pengujian dan porofolio asset untuk blockchain menggunakan (Etherum Virtual Machine – EVM) [1], [2], [4].

1.2.3 Ganache

Ganache merupakan blockchain pribadi dari etherum untuk membangun dan menguji pengembangan seperti Dapps dan Smart Contracts menggunakan Solidity [1], [4], [5].

1.2.4 Flask

Flask merupakan framework yang tergolong sebagai jenis microframework dan berfungsi sebagai kerangka kerja aplikasi dan tampilan web. Penggunaan Flask dan Python dapat memudahkan developer untuk membuat web yang lebih terstruktur. Flask termasuk dalam jenis microframework karena penggunaannya tidak memerlukan alat atau pustaka tertentu. Untuk menambahkan fitur dan komponen yang sudah disediakan oleh pihak ketiga, seperti validasi formulir, pengelolaan upload, dan database, Flask dapat menggunakan ekstensi [6], [7].

1.2.5 SHA-256

Untuk keamanan kriptografi, algoritma hash aman 256-bit, atau SHA-256, biasanya digunakan. Hash kriptografi dibuat oleh algoritma yang tidak dapat dibaca. Peluang bahwa dua nilai akan menghasilkan hash yang sama berbanding lurus dengan kemungkinan hash. Karena fungsi hash satu arah pada dasarnya tidak dapat mengembalikan pesan ke bentuk aslinya. Untuk alasan inilah fungsi hash selalu digunakan dalam penyandian password [8].

SHA-256 mengubah pesan masukan ke dalam message digest (MD) sebesar 256-bit. Pesan yang panjangnya kurang atau lebih dari 264-bit harus dioperasikan oleh kelompok 512-bit dan kemudian menjadi MD dengan panjang tetap, yaitu 256 bit [9].

1.2.6 Web3js

Web3.js merupakan sekumpulun library Javascript yang dapat terintegrasi dengan node Ethereum lokal menggunakan HTTP, WebSocket, atau IPC. Dengan API Web3.js, front-end bisa berinteraksi dengan *Smart Contract*. Web3.js merupakan jembatan bagi JSON RPC Ethereum menggunakan Bahasa pemrograman Javascript sebagai antarmukanya, yang memungkinkannya untuk digunakan secara instan dalam aplikasi web, sebab JavaScript didukung baik oleh hampir seluruh browser. Penggunaan Web3.js juga dapat digunakan sebagai penghubung ke jaringan Ethereum dengan menggunakan node Ethereum yang dapat diakses melalui HTTP [10].

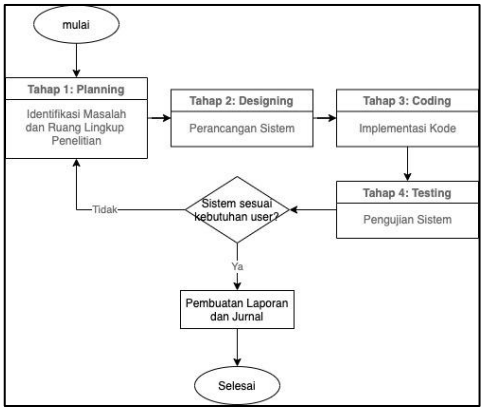
1.3 Penelitian Terkait

Penelitian sebelumnya yang berjudul “Certificate Verification using Blockchain and Generation of Transcript”, digunakan sebagai acuan dalam pembuatan aplikasi verifikasi sertifikat berbasis website menggunakan teknologi blockchain. Pada

penelitian ini bertujuan untuk memudahkan para validator dalam memvalidasi sertifikat para siswa di India [2].
Berdasarkan penelitian terkait penulis bersama tim merancang aplikasi validasi sertifikat menggunakan blockchain berbasis website. Keamanan dalam aplikasi ini akan lebih terjamin dibandingkan dengan aplikasi yang tidak menggunakan teknologi blockchain dikarenakan apapun yang tercatat tidak dapat dirusak atau diubah kembali dan meskipun dapat dirusak atau diubah akan sangat mudah dan cepat untuk mengetahui ada data yang telah berubah. Dalam aplikasi yang kami bangun memiliki beberapa fungsi yaitu, truffle, ganache, flask, sha256 file, ajax, bootstrap, web3js.

2. Metode Penelitian

Metode yang dilakukan dalam penelitiain ini seperti yang tertera pada gambar 1 *flowchart* atau alur penelitian:



Gambar 1. Diagram alur atau *flowchart* penelitian

2.1 Planning

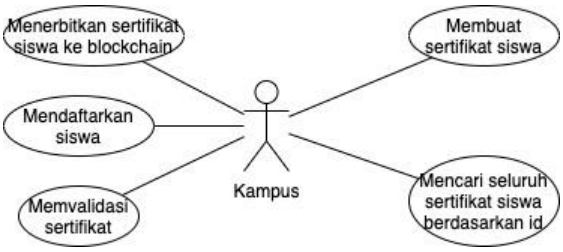
Tahap awal pada penelitan ini adalah tahap *planning* yakni untuk merencanakan apa saja kebutuhan untuk membuat aplikasi ini. Pada tahap ini akan didefinisikan lingkup serta batasan penelitian, dan pembuatan *user stories* untuk menspesifikasikan fungsi apa saja yang dapat bermanfaat bagi pengguna.

2.2 Designing

Tahap selanjutnya setelah perencanaan kebutuhan sistem adalah desain, pada tahap ini peneliti mulai melakukan perancangan desain aplikasi melalui pendekatan UML antara lain *use case diagram*, dan *activity diagram.*

2.2.1 Use case diagram

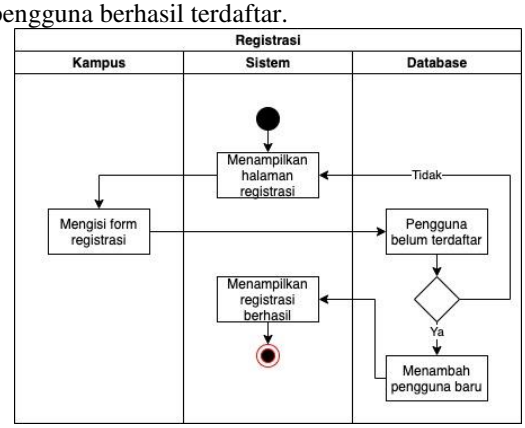
Use Case dari Aplikasi CerTion yang telah dibuat terdapat satu aktor saja yakni kampus atau bisa disebut sebagai admin. Kampus dapat mendaftarkan siswa, membuat sertifikat siswa, menerbitkan setifikat siswa ke dalam blockchain, mencari seluruh sertifikas siswa berdasarkan id siswa yang telah terdaftar dan mevalidasi sertifikas tersebut.



Gambar 2. Use Case Diagram CerTion

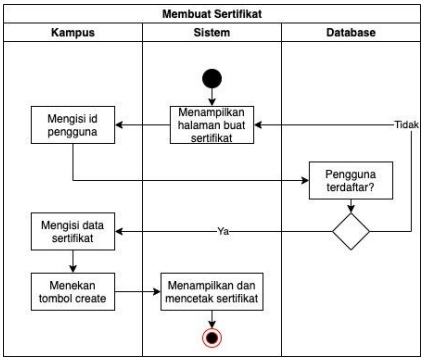
2.2.2 Activity Diagram

Gambar 3 merupakan *activity* diagram dari proses registrasi. Berdasarkan *activity* diagram tersebut, pada proses registrasi sistem akan menampilkan halaman registrasi yang kemudian akan diisi oleh pihak kampus sesuai dengan data siswanya, apabila pengguna telah terdaftar maka sistem akan menampilkan akun telah terdaftar dan menampilkan kembali halaman registrasi, jika pengguna belum terdaftar maka akun pengguna akan terdaftar ke dalam *database* kemudian sistem akan menampilkan pengguna berhasil terdaftar.



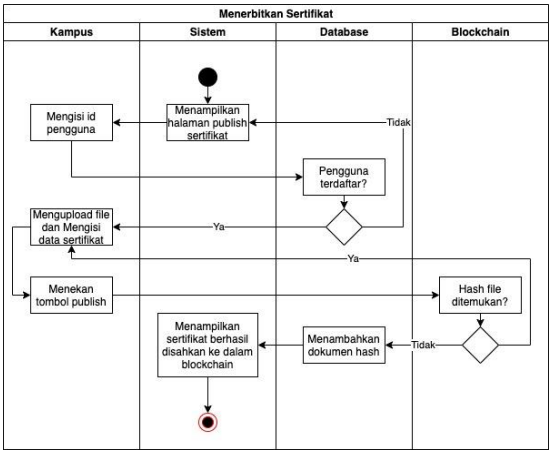
Gambar 3. Activity Diagram Registrasi

Gambar 4 merupakan *activity* diagram dari proses membuat sertifikat. Berdasarkan *activity* diagram tersebut, pada proses membuat sertifikat sistem akan menampilkan halaman buat sertifikat, kemudian pihak kampus akan mengisi id pengguna, jika pengguna belum terdaftar maka akan kembali menampilkan halaman buat sertifikat, jika telah terdaftar kampus akan mengisi data sertifikat dan menekan tombol *create* setelah semua data telah terisi, maka sistem akan menampilkan dan mencetak sertifikat.



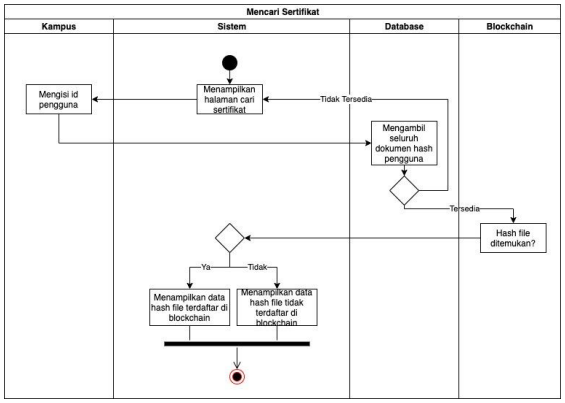
Gambar 4. Activity Diagram Membuat Sertifikat

Gambar 5 merupakan *activity* diagram dari proses menerbitkan sertifikat. Berdasarkan *activity* diagram tersebut, pada proses menerbitkan sertifikat sistem akan menampilkan halaman *publish* sertifikat, kemudian pihak kampus akan mengisi id pengguna, jika pengguna belum terdaftar maka akan kembali menampilkan halaman *publish* sertifikat, jika telah terdaftar kampus akan mengunggah dan mengisi data sertifikat lalu menekan tombol *publish* setelah semua data telah terisi, maka data tersebut akan dicek dalam *blockchain*, jika *hash file* ditemukan mana akan kembali ke halaman mengunggah file dan mengisi data, apabila *hash file* tidak ditemukan maka dokumen *hash* akan ditambahkan ke dalam *database* lalu sistem akan menampilkan sertifikat berhasil disahkan ke dalam *blockchain*.



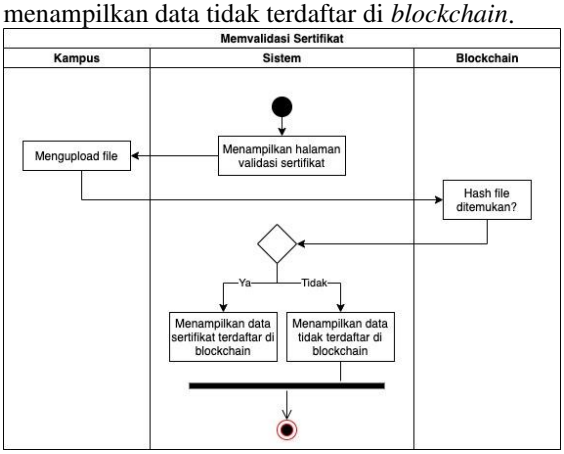
Gambar 5. Activity Diagram Menerbitkan Sertifikat

Gambar 6 merupakan *activity* diagram dari proses mencari sertifikat. Berdasarkan *activity* diagram tersebut, pada proses mencari sertifikat sistem akan menampilkan halaman cari sertifikat kemudian pihak kampus akan mengisi id pengguna, lalu mengambil seluruh dokumen *hash* pengguna jika tidak ditemukan maka akan kembali ke halaman cari sertifikat, apabila ditemukan maka akan dicek *hash file*nya, jika ditemukan akan menampilkan data *hash* file terdaftar di *blockchain*, jika tidak ditemukan maka menampilkan data *hash file* tidak terdaftar di *blockchain*.



Gambar 6. Activity Diagram Mencari Sertifikat

Gambar 7 merupakan *activity* diagram dari proses mamvalidasi sertifikat. Berdasarkan *activity* diagram tersebut, pada proses memvalidasi sertifikat sistem akan menampilkan halaman validasi sertifikat kemudian pihak kampus akan mengunggah *file,* kemudian akan dicek *hash file*nya, jika ditemukan maka akan menampilkan data sertifikat terdaftar di *blockchain*, jika tidak ditemukan maka akan menampilkan data tidak terdaftar di *blockchain*.



Gambar 6. Activity Diagram Memvalidasi Sertifikat

2.3 **Coding**

Pada tahap ini dilakukan implementasi *planning* dan *designing* ke dalam bentuk *coding*. Dimana proses pembuatan aplikasi ini akan dibuat sesuai dengan fungsionalitasnya berdasarkan *user stories* yang telah didapat.

2.4 **Testing**

Pada tahap *testing* fokus pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak mengalami *error* saat digunakan pengguna dan sesuai dengan proses bisnis pengguna atau pelanggan.

3. **Hasil dan Pembahasan**

3.1 **Implementasi Kode**

Certify.sol contracts

```
pragma solidity ⩾ 0.5.0 < 0.7.0;

contract Certify {

  struct Record {
      uint mineTime;
      uint blockNumber;
      string instituteName;
      string recipientName;
      string courseName;
      string marks;
      string dateOfCompletion;
  }

  mapping (bytes32 ⇒ Record) private docHashes;

  // function that is used to store records on the blockchain
  function addDocHash(bytes32 hash, string memory insti, string memory reci,
  string memory course, string memory grade, string memory doc ) public {
    Record memory newRecord = Record(block.timestamp, block.number, insti,
    reci, course, grade,doc);
    docHashes[hash] = newRecord;
  }

  // verify weather a hash exists on the blockchain and retrieve the
  corresponding record
  function findDocHash(bytes32 hash) public view returns(uint, string memory,
  string memory, string memory, string memory, string memory) {
    return (docHashes[hash].blockNumber, docHashes[hash].instituteName,
    docHashes[hash].recipientName,
        docHashes[hash].courseName, docHashes[hash].marks, docHashes[hash].
        dateOfCompletion);
  }
}
```

Gambar 7. Kelas Smart Contract Certify

File 'Certify.sol' merupakan sebuah *smart contract* menggunakan bahasa solidity. *Solidity* adalah bahasa tingkat tinggi berorientasi objek untuk mengimplementasikan *smart contract*. *Smart contract* adalah program yang mengatur perilaku akun Ethereum. Didalam sini terdapat beberapa fungsi, *struct* dan *mapping* sebagai berikut.

1. *Struct Record*

```
struct Record {
    uint mineTime;
    uint blockNumber;
    string instituteName;
    string recipientName;
    string courseName;
    string marks;
    string dateOfCompletion;
}
```

Gambar 8. Kode Struct Record

Struct adalah tipe yang ditentukan khusus yang dapat mengelompokkan beberapa variabel. *Struct* ini mengelompokkan variabel *mineTime* dengan tipe *unsigned integer*, *blockNumber* dengan tipe *unsigned integer*, *instituteName* dengan tipe *string*, *recipientName* dengan tipe *string*, *courseName* dengan tipe *string*, *marks* dengan tipe *string*, dan *dateOfCompletion* dengan tipe *string*.

Struct ini digunakan untuk menyimpan kumpulan *value* sertifikat pada *blockchain* di dalam *mapping*.

2. *Mapping docHashes*

```
mapping (bytes32 => Record) private docHashes;
```

Gambar 9. Kode Mapping docHashes

Mapping docHashes menyimpan key value. Key dari *docHashes* bertipe integer dengan value yaitu *struct Record*. *Mapping* ini digunakan untuk menyimpan kumpulan data sertifikat pada *blockchain*.

3. *Function addDocHash*

```
// function that is used to store records on the blockchain
function addDocHash(bytes32 hash, string memory insti, string memory reci,
string memory course, string memory grade, string memory doc ) public {
    Record memory newRecord = Record(block.timestamp, block.number, insti,
    reci, course, grade,doc);
    docHashes[hash] = newRecord;
}
```

Gambar 10. Fungsi addDocHash

Function addDocHash merupakan sebuah *fungsi public* untuk menambahkan dokumen sertifikat pada jaringan *blockchain*, dengan memasukkan parameter *hash* berupa *bytes32*, insti, reci, course, grade, dan doc berupa *string* yang berada pada *memory*. Ketika menambahkan dokumen sertifikat maka akan diinisialisasi *mapping* daftarTugas pada *key* dengan index dari hash, dan *value* yaitu *struct Record*.

4. *Function findDocHash*

```
// verify weather a hash exists on the blockchain and retrieve the
corresponding record
function findDocHash(bytes32 hash) public view returns(uint, string memory,
string memory, string memory, string memory) {
    return (docHashes[hash].blockNumber, docHashes[hash].instituteName,
    docHashes[hash].recipientName,
        docHashes[hash].courseName, docHashes[hash].marks, docHashes[hash].
        dateOfCompletion);
}
```

Gambar 11. Fungsi findDocHash

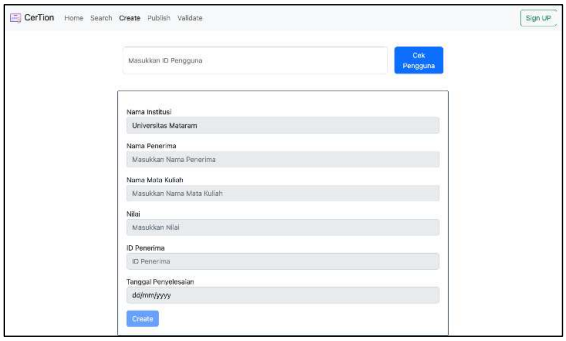
Function findDocHash merupakan sebuah *fungsi public* untuk mencari dokumen sertifikat pada jaringan *blockchain*, dengan memasukkan parameter *hash* berupa *bytes32*. Fungsi ini akan menegembalikan data sertifikat pada index *hash*.

3.2 **Implementasi Sistem**

Gambar 12 merupakan halaman *home*, terdapat beberapa fitur utama yang tersedia pada aplikasi ini diantaranya adalah membuat sertifikat pengguna dengan format PDF, memvalidasi sertifikat berdasarkan hash file yang terdaftar pada *blockchain*, menerbitkan sertifikat pengguna ke dalam *blockchain* dan melakukan pencarian seluruh sertifikat pengguna berdasarkan ID pengguna.

Gambar 12. Halaman Home

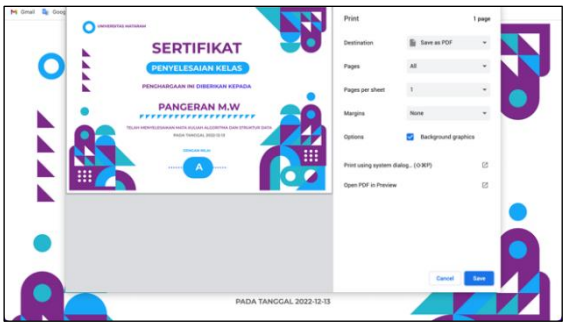
Gambar 13 merupakan halaman untuk membuat sertifikat dengan cara memasukkan ID pengguna terlebih dahulu, jika terdaftar maka *form fieldset* untuk mengisi sertifikat akan aktif (tidak *disabled*). Pihak kampus perlu mengisi semua *form fieldset* yang tersedia seperti nama institusi, nama penerima, nama mata kuliah, nilai, dan tanggal penyelesaian.



Gambar 13. Halaman Create Sertifikat

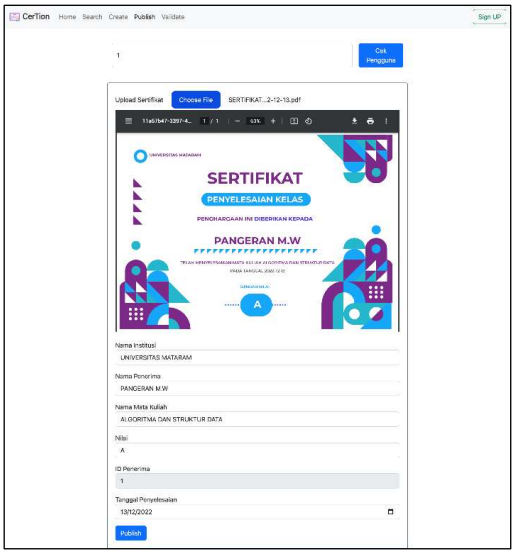
Jika sudah terisi maka akan sertifikat akan dibuat sesuai data dan diarahkan ke halaman cetak sertifikat sesuai pada Gambar 14, untuk menyimpan file

sertifikat yang akan digunakan ketika menerbitkannya ke *blockchain*.



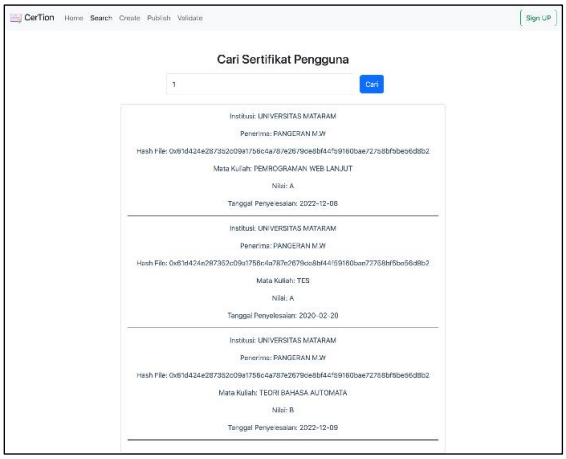
Gambar 14. Halaman Cetak Sertifikat

Gambar 15 merupakan halaman untuk menerbitkan sertifikat dengan cara memasukkan ID pengguna terlebih dahulu, jika terdaftar maka *form fieldset* untuk menerbitkan sertifikat akan aktif (tidak *disabled*). Pihak kampus perlu mengisi semua *form fieldset* yang tersedia seperti *file* sertifikat, nama institusi, nama penerima, nama mata kuliah, nilai, dan tanggal penyelesaiain.



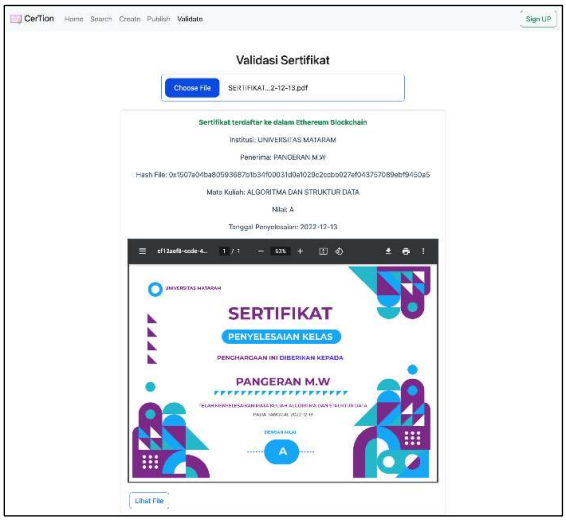
Gambar 15. Halaman Publish Sertifikat

Gambar 16 merupakan halaman untuk mencari seluruh sertifikat berdasarkan ID pengguna dengan cara memasukkan ID pengguna terlebih dahulu, jika terdaftar dan memiliki sertifikat maka akan ditampilkan seluruh sertifikat yang telah terdaftar pada *blockchain* seperti nama institusi, nama penerima, nama mata kuliah, nilai, dan tanggal penyelesaian dan *hash file*.



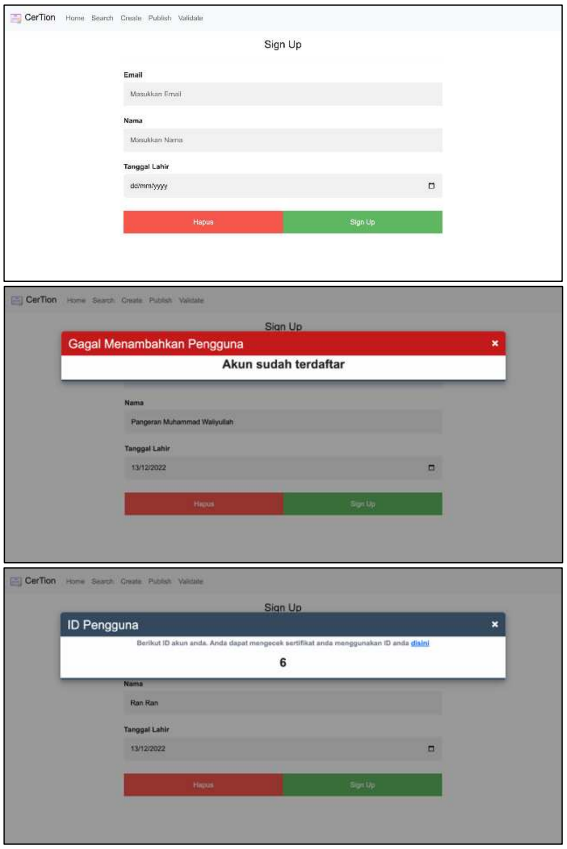
Gambar 16. Halaman Search Sertifikat

Gambar 17 merupakan halaman untuk memvalidasi sertifikat dengan cara meng-*upload file* sertifikat berupa PDF terlebih dahulu, jika terdaftar pada *blockchain* maka akan ditampilkan tulisan hijau yaitu "sertifikat terdaftar ke dalam *ethereum blockchain*" dan info sertifikat seperti nama institusi, nama penerima, nama mata kuliah, nilai, dan tanggal penyelesaian dan *hash file* serta *file* yang di-*upload*.



Gambar 17. Halaman Validasi Sertifikat

Gambar 18 merupakan halaman untuk membuat akun dengan cara memasukkan data yang diperlukan. Ketika *email* belum didaftarkan maka akun berhasil dibuat, namun jika sebaliknya maka akan *error*.



Gambar 18. Halaman SignUp

3.3 **Pengujian**

Pengujian sistem menggunakan metode *blackbox testing* dengan menguji fungsionalitas fitur. Pengujian ini dilakukan untuk mengecek apakah input dan output dari aplikasi sudah sesuai dan berjalan dengan baik atau tidak, tanpa mengetahui kode yang ada di dalamnya.

Tabel 1. Pengujian *Black Box Testing*

No	Skenario	*Test Case*	Harapan	Hasil
1.	Buat akun	Membuat akun pengguna baru	Dapat membuat akun pengguna baru	Berhasil
2.	Buat sertifikat	Membuat sertifikat pengguna	Dapat membuat sertifikat pengguna	Berhasil
3.	Publikasi sertifikat	Mempublikasikan sertifikat pengguna	Dapat mempublikasikan sertifikat pengguna	Berhasil
4.	Cari sertifikat	Mencari sertifikat pengguna yang telah dipublikasi	Dapat mencari sertifikat pengguna yang telah dipublikasikan	Berhasil
5.	Validasi sertifikat	Memvalidasi sertifikat pengguna yang telah dimasukkan ke blockhain	Dapat memvalidasi sertifikat yang telah dimasukkan di blockchain	Berhasil

Berdasarkan tabel pengujian di atas dapat diambil kesimpulan bahwa semua fitur yang ada pada aplikasi berhasil berjalan dengan baik.

4. **Kesimpulan**

Adapun kesimpulan dari projek tugas besar ini yaitu:

1. Telah berhasil dibuat aplikasi yang dapat membantu dalam memvalidasi sertifikat para siswa/mahasiswa.
2. Teknologi blockchain dapat membantu sistem peminjaan barang yang tidak dapat diubah, dihapus, serta dapat menyimpan data yang transparan dan dapat diakses oleh public.
3. Fitur - fitur yang terdapat pada aplikasi validasi sertifikat dengan blockchain yaitu membuat akun pengguna, membuat sertifikat, mempublikasi sertifikat, mencari sertifikat, dan memvalidasi sertifikat.

Daftar Pustaka

[1] K. Bhosale, K. Akbarabbas, J. Deepak, and A. Sankhe, “Blockchain based Secure Data Storage,” *International Research Journal of Engineering and Technology*, vol. 06, no. 03, pp. 1–4, 2019, [Online]. Available: www.irjet.net

[2] R. S. Lamkoti, D. Maji, A. Bharati Gondhalekar, and H. Shetty, “Certificate Verification using Blockchain and Generation of Transcript,” *India*, vol. 10, no. 03, pp. 539–544, 2021, [Online]. Available: https://www.ijert.org/research/certificate-verification-using-blockchain-and-generation-of-transcript-IJERTV10IS030260.pdf

[3] A. S. Mustafa, M. A. Al-Mashhadani, S. A. Jasim, A. M. Shantaf, and M. M. Hamdi, “Blockchain in fifth-generation network and beyond: a survey,” *Bulletin of Electrical Engineering and Informatics; Vol 11, No 3: June 2022*, 2022, doi: 10.11591/eei.v11i3.3209.

[4] L. Yan, L. Ge, Z. Wang, G. Zhang, J. Xu, and Z. Hu, "Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment," *Journal of Cloud Computing*, vol. 12, no. 1, p. 61, 2023, doi: 10.1186/s13677-023-00444-4.

[5] T. Rama Reddy, P. V. G. D. Prasad Reddy, R. Srinivas, Ch. V Raghavendran, R. V. S. Lalitha, and B. Annapurna, "Proposing a reliable method of securing and verifying the credentials of graduates through blockchain," *EURASIP J Inf Secur*, vol. 2021, no. 1, p. 7, 2021, doi: 10.1186/s13635-021-00122-5.

[6] F. A. Aslam, H. N. M. J. M. M. M. M. A. Gulamgaus, and P. P. S. Lokhande, "Efficient Way Of Web Development Using Python And Flask," *International Journal of Advanced Research in Computer Science*, vol. 6, no. 2, pp. 54–57, 2015, [Online]. Available: www.ijarcs.info

[7] A. Jain and S. Soni, "MarbleBot: A Conversational Recommender and Assistance Chatbot for Marble Selection Based on Dialogflow," in *2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM)*, 2023, pp. 1–6. doi: 10.1109/ELEXCOM58812.2023.10370445.

[8] R. L. Quilala and T. F. G. Quilala, "Improved MSHA-1 algorithm with mixing method," *Bulletin of Electrical Engineering and Informatics; Vol 10, No 4: August 2021*, 2021, doi: 10.11591/eei.v10i4.2366.

[9] C. P. Sukhbir and S. S. Manivannan, "Crypto Currencies for Digital Currency Using Cipher Text and SHA256," *Journal of Advanced Research in Dynamical and Control Systems*, pp. 530–534, 2017.

[10] L. Song, X. Ju, Z. Zhu, and M. Li, "An access control model for the Internet of Things based on zero-knowledge token and blockchain," *EURASIP J Wirel Commun Netw*, vol. 2021, no. 1, 2021, doi: 10.1186/s13638-021-01986-4.